Storage QoS: a bit of our work + comments on open challenges

Greg Ganger
(for many)

PARALLEL DATA LABORATORY

Carnegie Mellon University

Goal: shared storage with per-user Storage Q

- User specifies goals, system achieves them
- Sharing allows common namespace
- Sharing allows common provision+use of spare
  - including bursty usage

But, storage QoS is quite difficult (given “goals

- Device performance varies wildly
  - across devices, workloads, and time
- Inter-user interference can kill storage performan
- Scale: coordinating I/O scheduling across nodes

Across workloads

- we know this: Random vs. sequential

Across devices (even of same make/model)

- by design, no two disks are identical [Krevat’11]

Across time

- Modern devices have all sorts of “random” performance effects (e.g., background activities)

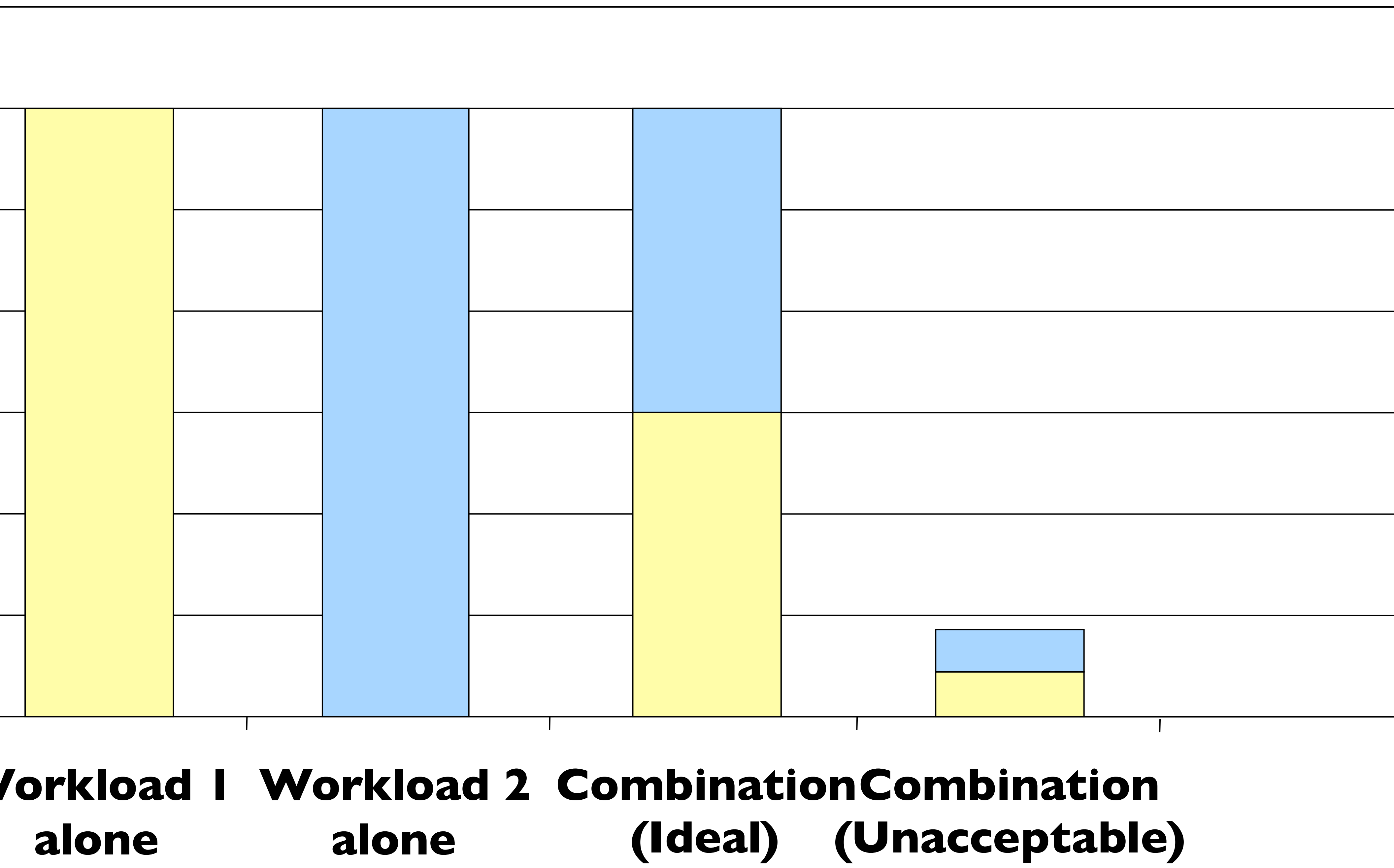
An interleaved workload is different from either

- e.g., two sequentials can look “random”
- e.g., one may evict the others’ cache blocks

Can result in dramatic performance degradatio

And, worse for storage QoS, unpredictable per

- performance for workload A depends on B’s activ
- QoS control loops hate unpredictable changes

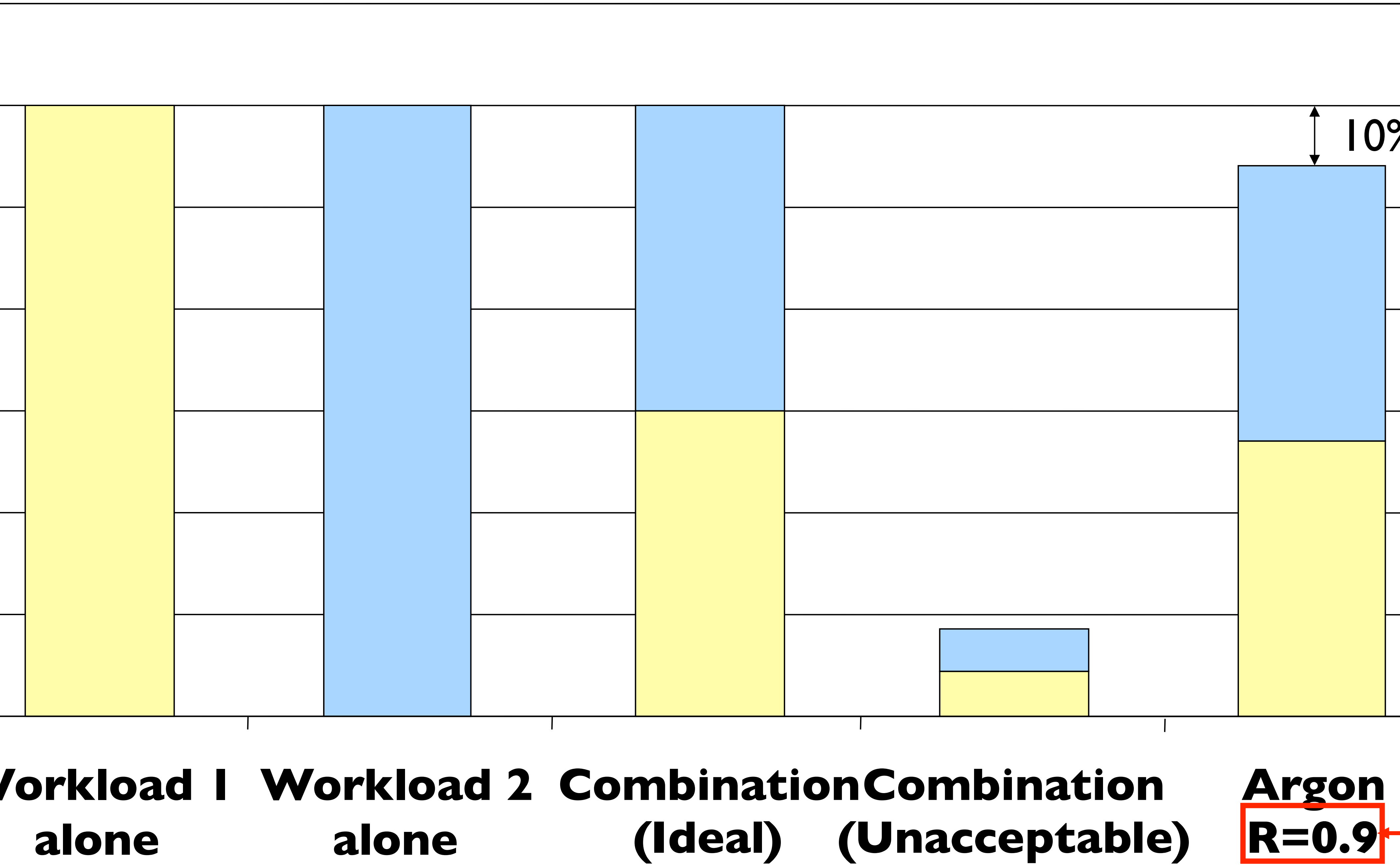

Workload 1 alone
Workload 2 alone
Combination (Ideal)
Combination (Unacceptable)

Ideal: each of n workloads on a server

- gets at least some explicit fraction of server “time
  - e.g., $1/n$ or a chosen proportion
- does not lose efficiency because of sharing
  - i.e., at least as efficient as when running alor

Practical goal: an explicit “R-value” [Wachs07

- a configurable lower bound on efficiency
  - measured as throughput relative to non-sharing
  - adjusted according to the fraction of server time


10%
Workload 1 alone
Workload 2 alone
Combination (Ideal)
Combination (Unacceptable)
Argon
R=0.9

Insulation (Argon) bounds the interference

- leaving the QoS control loop to select share size
- remove the "avoidable" QoS violations [Wachs'1
  - i.e., those resulting from interference
- … and some "unavoidable" ones
  - by exploiting slack wisely

"Unavoidable" ones that remain…

- e.g., workload changes its access patterns
- e.g., device performance changes unexpectedly

Data striped for performance (esp. bandwidth)

- each client req. translates to multiple server acce
- client req. is “done” when all accesses are done
  - so, overall req. waits for the slowest one

Must coordinate scheduling across servers

- can synchronize Argon quanta [Wachs’09]
- but…
  - what about inter-device variation?
  - what about “puzzle piece” data distributions?

any open questions/challeng

Goal specification

- sufficiently expressive while being usable & usefu
- poorly understood, even just for performance

Coordinating at scale

- when performance across devices differs (as it do
  - both consistently and intermittently
- when workloads vary across phases (as they do)
  - don’t want to idle entire storage cluster
- when not all data is striped in same way

DONE!

Greg.Ganger@cmu.edu

Director, Parallel Data Lab

PARALLEL DATA LAB

Carnegie Mellon Ur

rie Mellon